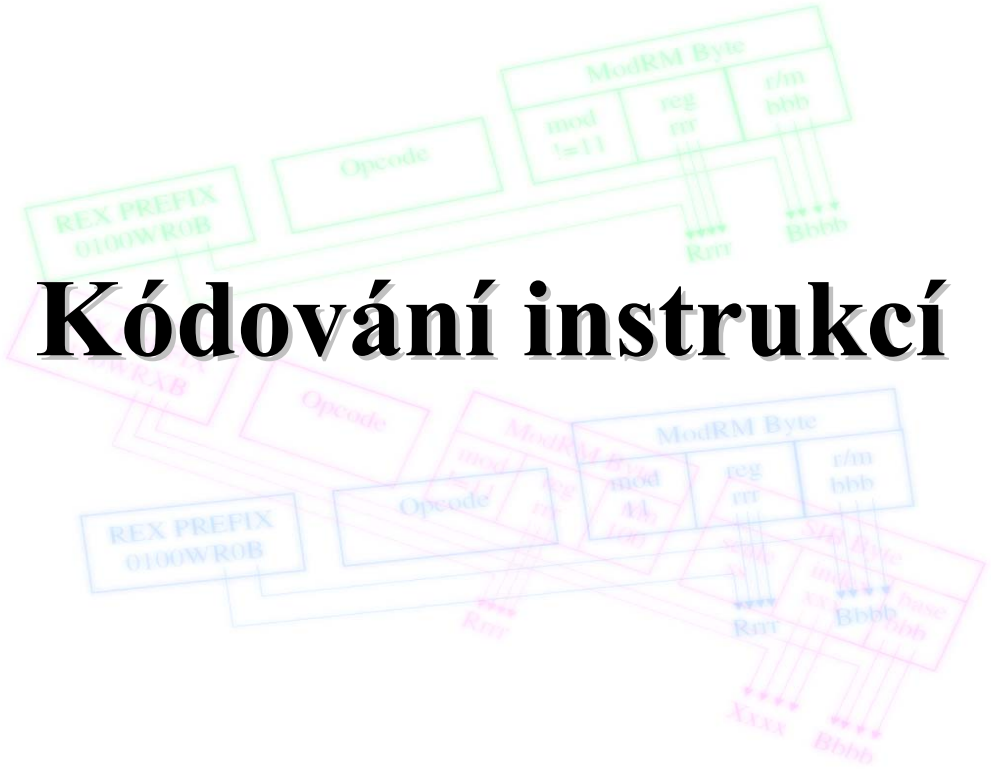
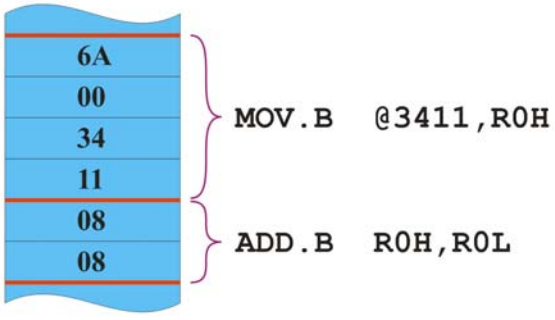


Kódování instrukcí

Úvod

Instrukce musí obsahovat:

- typ operace,
- adresu operandu (operandů),
- typ operandů,
- modifikátory adresy,
- modifikátory operace.



Důsledky použitého adresního modu

Adresní mod má vliv na kód instrukce:

- Přímý operand
  - Celý operand musí být uložen v instrukci – délka závisí na typu operandu (byte, word, dword).
- Registrový operand
  - Pro určení registru je obvykle zapotřebí 3 – 5 bitů.
- Přímá adresa
  - Délka adresy závisí na velikosti adresního prostoru (obvykle 16 – 32 bitů).
- Nepřímá adresa v registru
  - V instrukci je uloženo jen označení registru (cca 3 – 5 bitů).
- Bázová a indexová adresa
  - V instrukci musí být uloženo označení registru a offset.

Délka instrukcí

Podle způsobu kódování mohou být:

- Instrukce s pevnou délkou.
  - Všechny instrukce jsou stejně dlouhé (např. 32 bitů).
  - Časté u procesorů RISC.
- Instrukce s proměnnou délkou.
  - Jednotlivé instrukce jsou zakódované do různého počtu bytů v závislosti na počtu operandů, délce adresy nebo přímého operandu atd.
  - Časté u procesorů CISC.

Pevná a proměnná délka instrukce (1)

Pevná délka instrukce

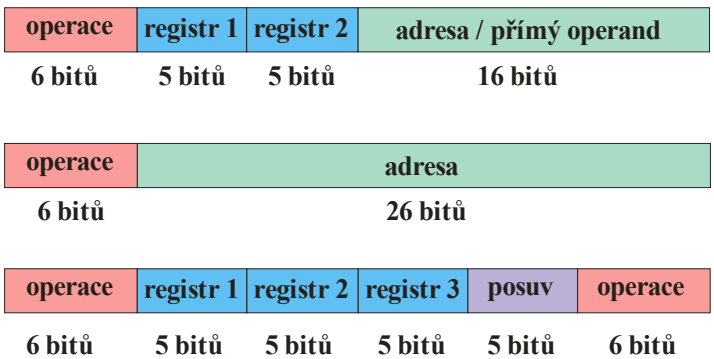
+ snadné a rychlé čtení a dekódování
- nízká efektivita kódování

Příklad: MIPS

- MIPS je procesor RISC s registrovou ISA (32 registrů).
- MIPS má 3 základní formáty instrukcí:
  - Formát I (Immediate).
  - Formát J (Jump).
  - Formát R (Register).
- Všechny instrukce mají délku 32 bitů.
- Adresování paměti je možné jen bázovou adresou (součet obsahu některého registru a offsetu v instrukci).

Instrukce s pevnou délkou (MIPS)

- MIPS má 3 základní formáty instrukcí:
  - Formát I (Immediate).
  - Formát J (Jump).
  - Formát R (Register).
- Všechny instrukce mají délku 32 bitů.



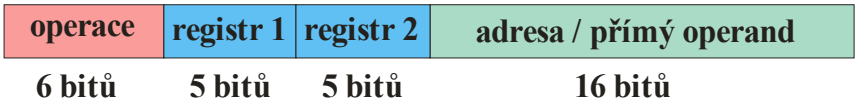
MIPS - Formát I

- Obsahuje přímý operand o délce 16 bitů.
  - Offset pro výpočet adresy (signed, tj. < - 32768; + 32767 >), např. instrukce

```
LW $1,-0x0200($2)     # $1 ← [$2-200]
```

  - Konstanta pro přímé operace, např. instrukce

```
ADDIU $1,$2,0x1234    # $1 ← $2 + 1234
```



MIPS - Formát J

- Formát pro skokové instrukce. Obsahuje 26 bitový offset pro cílovou adresu.
  - Offset se posouvá o 2 bity doleva, tj. skutečný rozsah offsetu je $< -2^{27}; +2^{27}-1 >$.
  - Příklad: instrukce `J lab_1` .

MIPS - Formát R

- Obsahuje pole pro určení 3 registrů (source, target, destination), pole pro délku posuvu a rozšířený operační kód.

  – Příklad:
  instrukce `ADD $1,$2,$3  # $1 ← $2 + $3` .

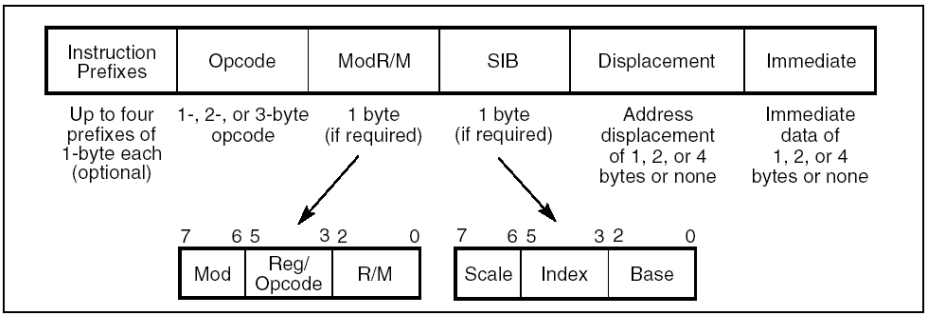
operace	registr 1	registr 2	registr 3	posuv	operace
6 bitů	5 bitů	5 bitů	5 bitů	5 bitů	6 bitů

Pevná a proměnná délka instrukce (2)

Proměnná délka instrukce

- obtížné čtení a dekódování
+ efektivnější kódování složitých instrukcí

Příklad: IA-32



IA-32 Instruction Format

Instrukční soubor IA-32

Aritmetické a logické operace

Operandy mohou být v registrech nebo v paměti.

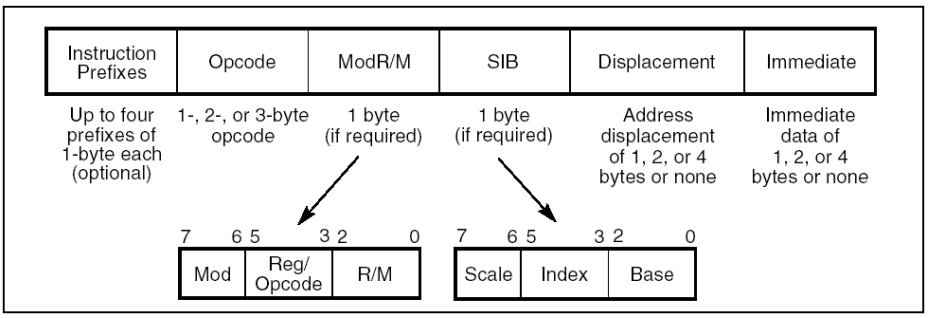
Dvouadresové instrukce.

např. **`ADD EAX,0x0100`** (EAX ← EAX + @0x0100).

Adresování paměti

Adresa může být např.:

`MOV EAX,EBX+[ECX*4]+0x0100` (EAX ← @(EBX + ECX*4 + 0x0100)).



IA-32 Instruction Format

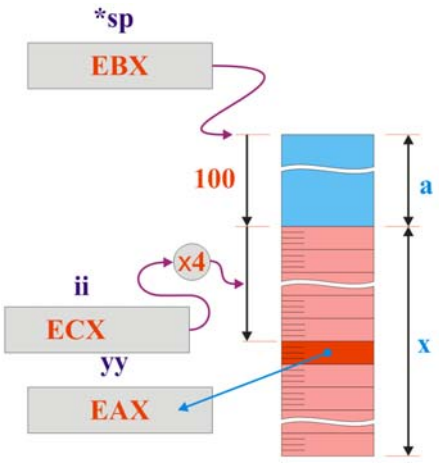
Instrukční soubor IA-32

Adresování paměti

Adresa může být např.:

`MOV EAX,EBX+[ECX*4]+0x0100` (EAX ← @(EBX + ECX*4 + 0x0100)).

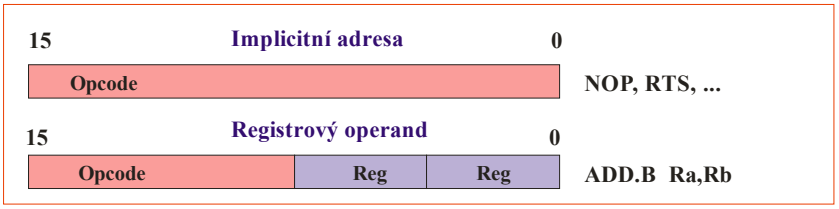
```
...
struct  str1{
        char    a[100];
        int     x[50];
};

struct str1     sss;
struct  str1    *sp;

sp = &sss;       /* sp = EBX */
ii = 20;         /* ii = ECX */
yy = sp->x[ii]; /* yy = EAX */
...
```



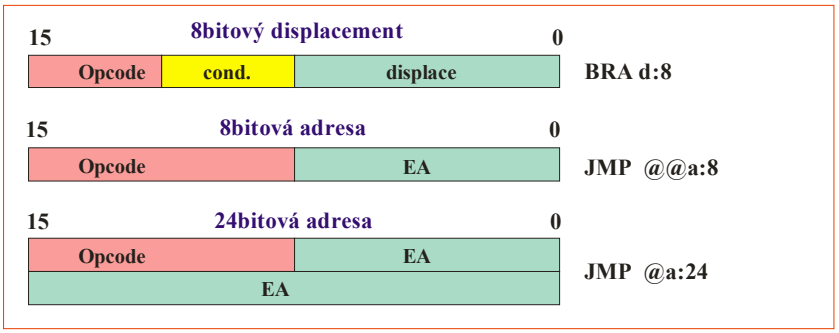
Kódování instrukcí procesoru H8S (1)

- Délka instrukcí je 2, 4, 6 nebo 8 bytů.
- Při adresování instrukcí (skoky atd.)je délka adresy 8 nebo 24 bitů.

Implicitní a registrový operand



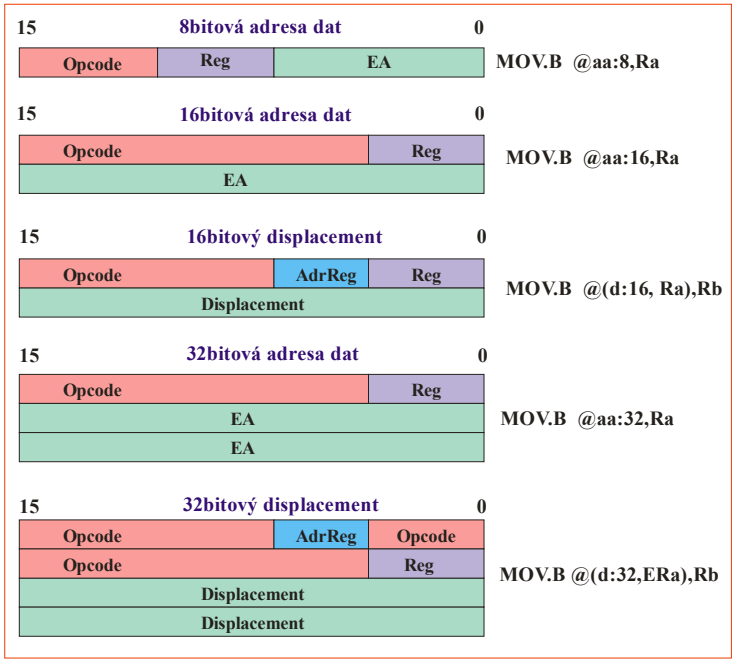
Adresování instrukcí



Kódování instrukcí procesoru H8S (2)

- Délka instrukcí je 2, 4, 6 nebo 8 bytů.
- Při adresování dat je délka adresy je 8, 16 nebo 32 bitů.

Adresování dat



Formát adresy dat procesoru H8S (2)

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)								
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	
MOV	MOV.B #xx:8,Rd	B	2								
	MOV.B Rs,Rd	B		2							
	MOV.B @ERs,Rd	B			2						
	MOV.B @(d:16, ERs), Rd	B				4					
	MOV.B @(d:32,ERs),Rd	B				8					
	MOV.B @ERs+,Rd	B					2				
	MOV.B @aa:8,Rd	B						2			
	MOV.B @aa:16,Rd	B						4			
	MOV.B @aa:32,Rd	B						6			
	MOV.B Rs,@ERd	B			2						
	MOV.B Rs,@(d:16,ERd)	B				4					
	MOV.B Rs,@(d:32,ERd)	B				8					
	MOV.B Rs,@–ERd	B					2				
	MOV.B Rs,@aa:8	B						2			
	MOV.B Rs,@aa:16	B						4			

Příklad: Instrukce **MOV.W**

MOV.W <EA>,Rd

Operand Format and Number of States Required for Execution

Addressing Mode	Mnemonic	Operands	Instruction Format								No. of States
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	
Immediate	MOV.W	#xx:16, Rd	7 9	0 rd	IMM						2
Register indirect	MOV.W	@ERs, Rd	6 9	0 ers rd							2
Register indirect with displace-ment	MOV.W	@(d:16, ERs), Rd	6 F	0 ers rd	disp						3
	MOV.W	@(d:32, ERs), Rd	7 8	0 ers 0	6 B	2 rd	disp				5
Register indirect with post-increment	MOV.W	@ERs+, Rd	6 D	0 ers rd							3
Absolute address	MOV.W	@aa:16, Rd	6 B	0 rd	abs						3
	MOV.W	@aa:32, Rd	6 B	2 rd	abs						4

Logický a fyzický adresní prostor

Logický adresní prostor

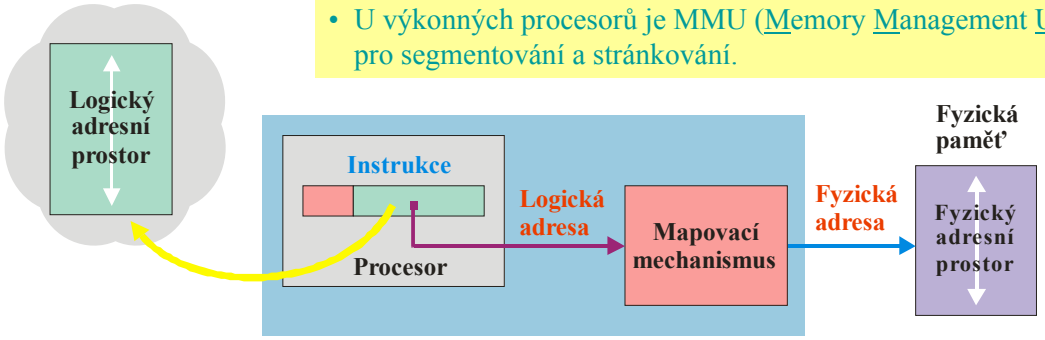
– Je určen rozsahem adres, které může procesor zpracovat (délka adres v instrukci + u některých procesorů modifikace adres segmentováním).

Fyzický adresní prostor

– Je určen velikostí fyzické paměti, se kterou může procesor pracovat (počet adresních vodičů).

Mapovací mechanismus:

- U jednoduchých procesorů není (logický a.p. = fyzický a.p.).
- U výkonných procesorů je MMU (Memory Management Unit) pro segmentování a stránkování.



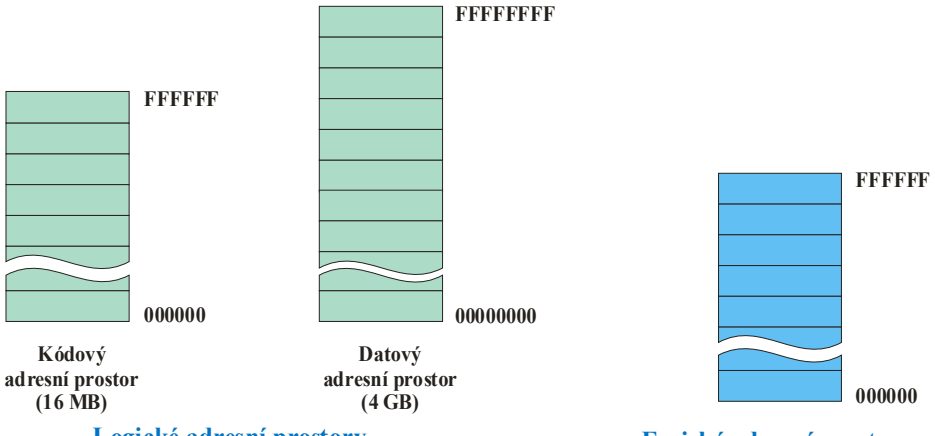
Adresní prostory H8S

Logický adresní prostor

– Je určen rozsahem adres, které může procesor zpracovat (délka adres v instrukci + u některých procesorů modifikace adres segmentováním).

Fyzický adresní prostor

– Je určen velikostí fyzické paměti, se kterou může procesor pracovat (počet adresních vodičů).

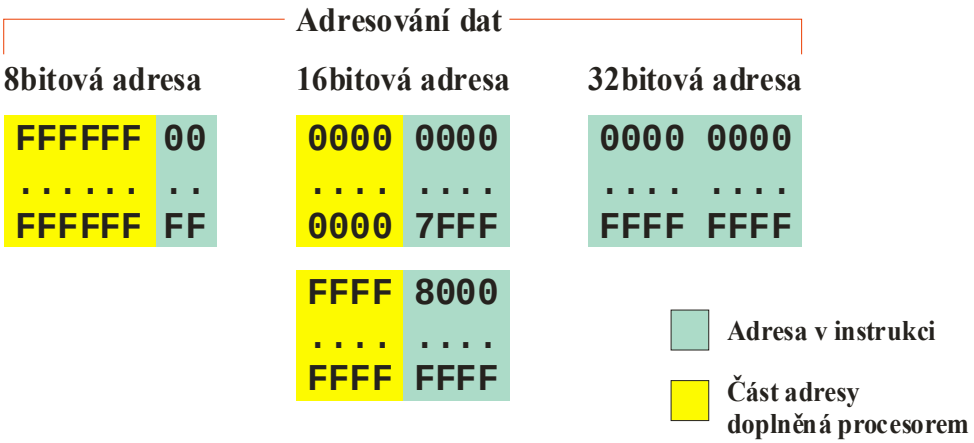


Formát adresy dat procesoru H8S

Logické adresování dat používá 32bitovou adresu.

- V instrukci lze použít 8, 16 nebo 32bitovou adresu.
- Vyšší bity adresy doplní procesor.

```
MOV.W @1A,R1        MOV.W @56AB,R1        MOV.W @12AB34CD,R1
```



Formát adresy instrukcí procesoru H8S

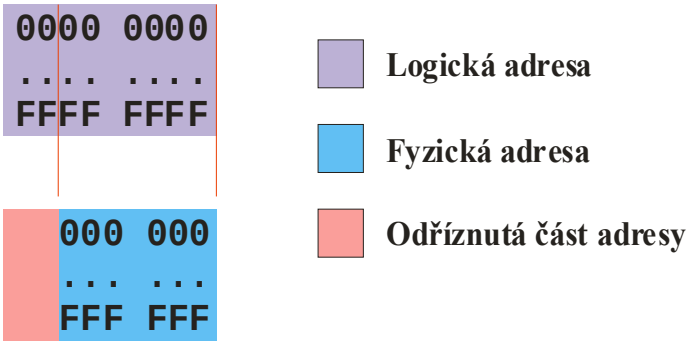
Logické adresování instrukcí používá 24bitovou adresu.

- V instrukci lze použít 8 nebo 24bitovou adresu.
- Vyšší adresní bity doplní procesor.


```
JMP @@1A            JMP @56AB12,R1
```

Převod logické adresy na fyzickou

- Logická adresa je 32bitová (data) nebo 24bitová (instrukce).
- Fyzická adresa je 24bitová.
- 8 nejvyšších bitů adresy dat se ignoruje.



Shrnutí